

# The list chromatic number of graphs with small clique number

Michael Molloy\*

February 1, 2017

## Abstract

We prove that every triangle-free graph with maximum degree  $\Delta$  has list chromatic number at most  $(1 + o(1))\frac{\Delta}{\ln \Delta}$ . This matches the best-known bound for graphs of girth at least 5. We also provide a new proof that for any  $r \geq 4$  every  $K_r$ -free graph has list-chromatic number at most  $200r\frac{\Delta \ln \ln \Delta}{\ln \Delta}$ .

## 1 Introduction

We provide new proofs of two results of Johansson. The proofs are much shorter and simpler, and obtain an improvement in the constant of the first result. We use entropy compression, a powerful new take on the Lovász Local Lemma. One of the contributions of this paper is to illustrate this approach.

The first result bounds the list chromatic number of a triangle-free graph. The list chromatic number of a graph  $G$  is the smallest  $q$  such that: for any assignment of colour-lists of size  $q$  to each vertex, it is possible to give each vertex a colour from its list and obtain a proper colouring. Johansson[14] proved that every triangle-free graph has list-chromatic number at most  $9\Delta/\log \Delta$  where  $\Delta$  is the maximum degree of the graph. The leading constant was improved to 4 in [22]. Here we obtain  $1 + o(1)$ :

**Theorem 1** *For every  $\epsilon > 0$  there exists  $\Delta_\epsilon$  such that every triangle-free graph  $G$  with maximum degree  $\Delta \geq \Delta_\epsilon$  has  $\chi_\ell(G) \leq (1 + \epsilon)\Delta/\ln \Delta$ .*

In other words: every triangle-free graph with maximum degree  $\Delta$  has list chromatic number at most  $(1 + o(1))\frac{\Delta}{\ln \Delta}$ .

This bound matches the best known bound for graphs of girth 4 [16], and indeed for any constant girth. The best known lower bound is  $\frac{1}{2}\frac{\Delta}{\ln \Delta}$  and comes from random  $\Delta$ -regular graphs. We remark that our proof yields an efficient randomized algorithm to find a colouring, and that this meets the so-called “algorithmic barrier” for random regular graphs. Finding an efficient algorithm to produce a  $(1 - \epsilon)\frac{\Delta}{\ln \Delta}$  colouring of a random  $\Delta$ -regular graph for some  $\epsilon > 0$  is a major open problem.

---

\*Dept of Computer Science, University of Toronto, molloy@cs.toronto.edu. Research supported by an NSERC Discovery Grant.

In a followup paper, Johansson[14] proved that for any constant  $r \geq 4$ , every  $K_r$ -free graph has list-chromatic number at most  $O(\Delta \ln \ln \Delta / \ln \Delta)$ . Here we match his bound, even when  $r$  grows with  $\Delta$ .

**Theorem 2** *For any  $r \geq 4$ , every  $K_r$ -free graph  $G$  with maximum degree  $\Delta$  has  $\chi_\ell(G) \leq 200r \frac{\Delta \ln \ln \Delta}{\ln \Delta}$ .*

Theorem 2 holds for any  $r$  but it is trivial unless  $r < \ln \Delta / 200 \ln \ln \Delta$ . Note also that this implies a bound on the chromatic number of  $H$ -free graphs for every fixed subgraph  $H$ , as an  $H$ -free graph is also  $K_{|H|}$ -free. (Here  $H$ -free means that there is no subgraph isomorphic to  $H$ ; the subgraph is not necessarily induced.)

These two results of Johansson were never published. His proof for triangle-free graphs was presented in [18] and his proof for  $K_r$ -free graphs was presented in [22].

It is a longstanding conjecture[5] that for constant  $r$ , every  $K_r$ -free graph has chromatic number  $O(\Delta / \log \Delta)$ . So we make no attempt to optimize the constant in Theorem 2. Thus far, we do not even know whether the independence number is large enough to support this conjecture. Prior to Johansson's work, Shearer[26, 27] proved that every triangle-free graph on  $n$  vertices has independence number at least  $(1 - o(1))n \ln \Delta / \Delta$  (see also [3]) and that every  $K_r$ -free graph has independence number at least  $\Omega(n \ln \Delta / \Delta \ln \ln \Delta)$ . His latter bound plays an important role in our proof of Theorem 2. Ajtai et al. conjectured that the  $\ln \ln \Delta$  term can be removed here [3].

Previous proofs of these, and similar results, used an iterative colouring procedure. In each iteration, one would colour some subset of the vertices, where each vertex received a random colour from its list. Every vertex that received the same colour as a neighbour would be uncoloured. (See [18] for a presentation of this technique.) One of the reasons for doing this is that the Local Lemma is much easier to apply when vertices are assigned colours independently. Entropy compression allows us to use Local Lemma like calculations for random colouring procedures where, roughly speaking, vertices are coloured one-at-a-time with colours not appearing on any neighbours.

This technique began with Moser's algorithm[19] which generated objects whose existence was guaranteed by the Local Lemma (see also [20] and see [28, 11] for good expositions of the technique). Subsequently, Grytczuk, Kozik and Micek[13] and Achlioptas and Iliopoulos[1] noted that his algorithm in fact can be applied to yield new existence results. Previous applications to graph colouring (eg. [10, 23, 2, 24, 7, 8, 12]) involved situations where, throughout the algorithm, each vertex is guaranteed to have a large number of available colours to choose from. That is not true in this paper since the degree of a vertex can be much higher than its list-size. The novelty we use here is to treat a vertex having a small number of available colours as a bad event.

## 1.1 Preliminary tools

We begin with a common version of the Local Lemma; see eg. Chapter 19 of [18].

**The Lovász Local Lemma**[9] *Let  $A_1, \dots, A_n$  be a set of random events, each with probability at most  $\frac{1}{4}$ . For each  $1 \leq i \leq n$  we have a subset  $\mathcal{D}_i$  of the events such that  $A_i$  is mutually independent of all other events outside of  $\mathcal{D}_i$ . If for each  $1 \leq i \leq n$  we have*

$$\sum_{j \in \mathcal{D}_i} \Pr(A_j) < \frac{1}{4}$$

then  $\Pr(\overline{A_1} \cap \dots \cap \overline{A_n}) > 0$ .

We say that boolean variables  $X_1, \dots, X_m$  are *negatively correlated* if

$$\text{for all } I \subseteq \{1, \dots, m\} : \quad \Pr(\wedge_{i \in I} X_i) \leq \prod_{i \in I} \Pr(X_i).$$

Panconesi and Srinivasan[21] noted that many Chernoff-type bounds on independent variables also hold on negatively correlated variables. We will use the following:

**Lemma 3** *Suppose  $X_1, \dots, X_m$  are negatively correlated boolean variables. Set  $X = \sum_{i=1}^m X_i$ . Then for any  $0 < t \leq \mathbf{E}(X)$ :*

$$\Pr(|X - \mathbf{E}(X)| > t) < 2e^{-t^2/3\mathbf{E}(X)}.$$

For independent variables, this follows from Theorem 2.3(b,c) in [17]. To adapt the proof so that it holds for negatively correlated variables, we only need one change. The proof for independent variables uses that for any  $h > 0$ :

$$\mathbf{E}(e^{hX}) = \mathbf{E}\left(\prod_{i=1}^m e^{hX_i}\right) = \prod_{i=1}^m \mathbf{E}(e^{hX_i}).$$

We replace this with

$$\mathbf{E}(e^{hX}) = \mathbf{E}\left(\prod_{i=1}^m e^{hX_i}\right) \leq \prod_{i=1}^m \mathbf{E}(e^{hX_i}).$$

For the remainder of the proof, see [17].

## 2 Triangle-free graphs

We have a set of colours  $\mathcal{C}$ . Each vertex  $v$  has a list  $\mathcal{L}_v \subseteq \mathcal{C}$  of size

$$|\mathcal{L}_v| = q := (1 + \epsilon) \frac{\Delta}{\ln \Delta}.$$

It suffices to prove Theorem 1 for small  $\epsilon$ ; in particular we will assume  $\epsilon < 1$ .

A partial list colouring  $\sigma$  is a colour assignment to a subset of the vertices, where the colours are drawn from their lists. Given any partial colouring,  $L_v \subseteq \mathcal{L}_v$  is defined to be the set of colours not appearing on any neighbours of  $v$ . We set

$$L = \Delta^{\epsilon/2}.$$

Note that if  $\Delta$  neighbours of  $v$  are each independently given a uniformly random colour from their lists, then the expected number of colours from  $\mathcal{L}_v$  that are not chosen for any neighbour is at least  $q(1 - 1/q)^\Delta \approx (1 + \epsilon)\Delta^{\frac{\epsilon}{1+\epsilon}} / \ln \Delta > L$ . So it is plausible that we can obtain a colouring in which every vertex has  $L_v > L$ . In fact we will prove that we can obtain a partial colouring with a substantial number of vertices coloured and with  $L_v > L$  for every  $v$ . From this, it will be straightforward to complete the colouring.

It will be convenient to treat Blank as a colour, and the uncoloured vertices are viewed as having been assigned this colour. Blank is the only colour that can be assigned to two neighbours.

Most of our work goes towards finding a partial list colouring with certain properties that make it easy to complete to a full colouring. Given a partial colouring  $\sigma$ , we define for each vertex  $v$  and colour  $c \neq \text{Blank}$ :

$$\begin{aligned} L_v &\text{ is the set of colours in } \mathcal{L}_v \text{ not appearing on } N_v, \text{ along with Blank.} \\ T_{v,c} &\text{ is the set of vertices } u \in N_v \text{ such that } \sigma(u) = \text{Blank and } c \in L_u \end{aligned}$$

Note that the preceding definition does not apply to  $T_{v,\text{Blank}}$ ; it will be convenient to set  $T_{v,\text{Blank}} = \emptyset$  for all  $v$ .

Given a partial colouring, we define the following two *flaws* for any vertex  $v$ :

$$\begin{aligned} B_v &\equiv |L_v| < L \\ Z_v &\equiv \sum_{c \in L_v} |T_{v,c}| > \frac{1}{10} L \times |L_v| \end{aligned}$$

We say  $v$  is the *vertex* of flaw  $f = B_v$  or  $Z_v$ , and we denote  $v(f) := v$ .

**Remark:** If we were content with proving the weaker bound of  $\chi_\ell(G) < (2 + o(1)) \frac{\Delta}{\ln \Delta}$  colours, then we could have defined  $Z_v$  to be a much simpler flaw, namely that  $v$  has at least  $L$  blank neighbours. We use that flaw in Section 3.

Our main goal is to find a partial colouring which has no flaws. The following proof that such a colouring can be completed to a proper colouring with no Blank vertices is essentially the proof of the main result in [25].

**Lemma 4** *Suppose we have a partial list colouring  $\sigma$  such that for every vertex  $v$ , neither  $B_v$  nor  $Z_v$  hold. Then we can colour the blank vertices to obtain a full list colouring.*

**Proof** We give each blank vertex  $v$  a uniformly chosen colour from  $L_v \setminus \text{Blank}$ . For any edge  $uv$  and colour  $c \in L_u \cap L_v$ ,  $c \neq \text{Blank}$  we define  $A_{uv,c}$  to be the event that  $u, v$  both receive  $c$ . Then  $\Pr(A_{uv,c}) = 1/(|L_u| - 1)(|L_v| - 1)$ . Furthermore,  $A_{uv,c}$  shares a vertex with at most  $\sum_{c' \in L_v} |T_{v,c'}| + \sum_{c' \in L_u} |T_{u,c'}|$  other events. The number of such events is at most  $\frac{1}{10} L(|L_v| + |L_u|)$  since  $Z_u, Z_v$  do not hold. It is straightforward to check that  $A_{uv,c}$  is mutually independent of all events with which it does not share a vertex (see eg. the Mutual Independence Principle in Chapter 4 of [18]). So our lemma follows from the Local Lemma as  $B_u, B_v$  do not hold and so

$$\frac{1}{(|L_u| - 1)(|L_v| - 1)} \times \frac{L(|L_v| + |L_u|)}{10} \leq \frac{L}{4.5(L - 1)} < \frac{1}{4},$$

for  $\Delta > 10^{2/\epsilon}$ . □

In the next section, we will present an algorithm to find a flaw-free colouring.

## 2.1 Our colouring algorithm

Consider a partial colouring  $\sigma$  and any flaw  $f$  of  $\sigma$ . We will use a recursive algorithm to correct  $f$ . Recall that every neighbourhood is an independent set, and so we recolour the vertices in a neighbourhood independently. We use the following ordering on the flaws:

Every  $B_v$  comes before every  $Z_u$ , and the  $B_v$ 's and  $Z_u$ 's are each ordered according to the labels of  $v, u$ . We use  $\text{dist}(w, v)$  to denote the distance from  $w$  to  $v$ ; i.e. the number of edges in a shortest  $w, v$ -path.

**FIX**( $f, \sigma$ )

(\*) Set  $v = v(f)$  and assign each  $u \in N_v$  a uniformly selected colour from  $L_u$ .

While there are any flaws  $B_w$  with  $\text{dist}(w, v) \leq 3$  or  $Z_w$  with  $\text{dist}(w, v) \leq 2$ :

Let  $g$  be the least such flaw and call **FIX** ( $g, \sigma'$ ) where  $\sigma'$  is the current colouring.

Return the current colouring.

**Remarks:**

(1) It is possible that  $f$  still holds after recolouring the neighbourhood of  $f$ , but then  $f$  itself would count as a flaw within distance 3 or 2 in the next line (but is not necessarily the least of those flaws). Note further that even if  $f$  does not hold after the recolouring, it is possible for future recolourings to bring  $f$  back and so **FIX** may be called again on  $f$  further down in the recursive calls.

(2) Because our graph is triangle-free, there are no edges in  $N_v$  and so we can recolour the vertices in  $N_v$  independently of each other. This greatly simplifies the analysis in Section 2.2.

(3) Because of our flaw ordering, whenever we call **FIX** ( $Z_u, \sigma$ ) we can assume that  $B_w$  does not hold for any  $w \in N_u$ . This is why we call **FIX** on flaws  $B_w$  with  $w$  up to distance three from  $v$  rather than two.

Next we note that if **FIX** terminates, then we have made progress in correcting the flaws.

**Observation 5** *In the colouring returned by **FIX**( $f, \sigma$ ):*

(a)  $f$  does not hold; and

(b) there are no flaws that did not hold in  $\sigma$ .

**Proof** Part (a) is true because we cannot exit the while loop if  $f$  holds. Part (b) is true because any new flaw  $f'$  must have arisen from recolouring a neighbour of  $v(f')$ ; i.e. during a call of **FIX** on some  $f''$  whose vertex is within distance at most two of  $v(f')$ . But we would not have exited the while loop of that call if  $f'$  still held.  $\square$

So we can obtain a flaw-free colouring by beginning with any partial colouring, eg. the all-Blank colouring, and then calling **FIX** at most once for each of the at most  $2n$  flaws of that colouring. Thus it suffices to prove that **FIX** terminates with positive probability; in fact, we will show that with high probability it terminates quickly (see the remark at the end of Subsection 2.3).

In the next subsection we prove that the proportion of colourings of  $N(v)$  for which  $f$  holds is at most  $\Delta^{-4}$ . In Subsection 2.3 we use that to show FIX terminates. Note that there are at most  $2\Delta^3$  flaws  $g$  which could appear in the while loop in FIX  $(f, \sigma)$ . Since  $2\Delta^3 \times \Delta^{-4} < \frac{1}{e}$  this feels like a Local Lemma computation. Entropy compression allows us to use such a computation in a procedure like FIX, which is more complicated than what we would typically apply the Local Lemma to; in particular note how quickly dependency spreads amongst the various flaws while running FIX.

## 2.2 Probability bounds

In this section, we prove the key bounds on the probability of our flaws.

**Setup for Lemma 6:** Each vertex  $u \in N_v$  has a list  $L_u$  containing Blank and perhaps other colours. We give  $u \in N_v$  a random colour from  $L_u$ , where the choices are made independently and uniformly. This assignment determines  $L_v, T_{v,c}$ .

**Lemma 6** (a)  $\Pr(|L_v| < L) < \Delta^{-4}$ .

(b)  $\Pr(\sum_{c \in L_v} |T_{v,c}| > \frac{1}{10}L \times |L_v|) < \Delta^{-4}$ .

### Remarks:

(1) This looks like an analysis of the probability that the recolouring in line (\*) produces another flaw on  $N_v$ . But we will actually apply it to count the number of choices for the flawed colouring that was on  $N_v$  at the beginning of the call to FIX.

(2) Kim's proof [16] for graphs of girth five was much simpler than Johansson's proof [14] for triangle-free graphs. The main reason was that if  $G$  has girth five then the neighbours of  $v$  have disjoint neighbourhoods (other than  $v$ ) which resulted in their lists being, in some sense, independent of each other. In a triangle-free graph with many 4-cycles, we could have two neighbours  $u_1, u_2$  of  $v$  whose neighbourhoods overlap a great deal and thus their lists would be highly dependent. Intuitively, it was clear that this should be helpful - if  $L_{u_1}$  and  $L_{u_2}$  are very similar then  $u_1, u_2$  would tend to get the same colour which would tend to increase the size of  $L_v$ . But, frustratingly, we did not know how to take advantage of this. In the current paper, the fact that dependencies between  $L_{u_1}, L_{u_2}$  do not hurt is captured by the fact that Lemma 6 holds for any set of lists on the neighbours of  $v$ .

**Proof** Part (a): For each colour  $c \in \mathcal{L}_v \setminus \text{Blank}$  we define:

$$\rho(c) = \sum_{u \in N_v : c \in L_u} \frac{1}{|L_u| - 1}.$$

Thus, since each  $L_u$  has  $|L_u| - 1$  non-Blank colours,

$$\sum_{c \in \mathcal{L}_v \setminus \text{Blank}} \rho(c) \leq \sum_{u \in N_v} \sum_{c \in L_u \setminus \text{Blank}} \frac{1}{|L_u| - 1} \leq \Delta.$$

If  $c \in L_u$  then  $|L_u| \geq 2$  and so we have  $1 - \frac{1}{|L_u|} > e^{-1/(|L_u|-1)}$ . We apply this inequality to obtain:

$$\mathbf{E}(|L_v|) = 1 + \sum_{c \in \mathcal{L}_v \setminus \text{Blank}} \prod_{u \in N_v : c \in L_u} \left(1 - \frac{1}{|L_u|}\right) > \sum_{c \in \mathcal{L}_v \setminus \text{Blank}} e^{-\rho(c)}. \quad (1)$$

By convexity of  $e^{-x}$  and  $\sum_{c \in \mathcal{L}_v \setminus \text{Blank}} \rho(c) < \Delta$  and recalling that  $|\mathcal{L}_v| = q = (1 + \epsilon)\Delta / \log \Delta$  we have

$$\mathbf{E}(|L_v|) > qe^{-\Delta/q} = \frac{(1 + \epsilon)\Delta}{\log \Delta} \times \Delta^{-\frac{1}{1+\epsilon}} > 2\Delta^{\epsilon/2} = 2L,$$

for  $\epsilon < 1$ .

To prove concentration, we set  $X_c$  to be the indicator variable that  $c \in L_v$ ; thus  $|L_v| = 1 + \sum_{c \in \mathcal{L}_v \setminus \text{Blank}} X_c$ . It is a simple exercise to check that for any set  $I$  of colours:

$$\Pr(\bigwedge_{c \in I} X_c) \leq \prod_{c \in I} \Pr(X_c).$$

Thus the variables  $X_c$  are negatively correlated and so Lemma 3 yields:

$$\Pr(|L_v| < \frac{1}{2}\mathbf{E}(|L_v|)) < 2e^{-\frac{1}{12}\mathbf{E}(|L_v|)} < 2e^{-\frac{1}{6}\Delta^{\epsilon/2}} < \Delta^{-4},$$

for  $\Delta$  sufficiently large in terms of  $\epsilon$ . This proves part (a).

*Part (b):* Let  $\Psi$  be the set of colours  $c \in L_v \setminus \text{Blank}$  with  $\rho(c) > \Delta^{\epsilon/4}$ . Using the same calculations as those for (1), but this time applying  $1 - \frac{1}{|L_u|} < e^{-1/|L_u|} < e^{-1/2(|L_u|-1)}$  for  $|L_u| \geq 2$ , the probability that  $L_v$  contains at least one colour from  $\Psi$  is at most

$$\mathbf{E}(|L_v \cap \Psi|) < \sum_{c \in \Psi} e^{-\frac{1}{2}\rho(c)} < qe^{-\frac{1}{2}\Delta^{\epsilon/4}} < \frac{1}{2}\Delta^{-4},$$

for  $\Delta$  sufficiently large in terms of  $\epsilon$ . For any  $c \notin \Psi$ :

$$\mathbf{E}(|T_{v,c}|) = \sum_{u: c \in L_u} \frac{1}{|L_u|} < \rho(c) \leq \Delta^{\epsilon/4}.$$

Since the choices of whether  $u \in T_{v,c}$ , i.e. whether  $u$  receives Blank, are made independently, Lemma 3 (or Theorem 2.3(b) of [17]) yields  $\Pr(|T_{v,c}| > \mathbf{E}(|T_{v,c}|) + \Delta^{\epsilon/4}) < e^{-\frac{3}{8}\Delta^{\epsilon/4}}$  and so the probability that there is at least one  $c \notin \Psi$  with  $|T_{v,c}| > 2\Delta^{\epsilon/4}$  is at most

$$qe^{-\frac{3}{8}\Delta^{\epsilon/4}} < \frac{1}{2}\Delta^{-4},$$

for sufficiently large  $\Delta$ . So with probability at least  $1 - \Delta^{-4}$  we have

$$\sum_{c \in L_v \setminus \text{Blank}} |T_{v,c}| = \sum_{c \in L_v \setminus \Psi} |T_{v,c}| \leq 2\Delta^{\epsilon/4}|L_v| < L \times |L_v|.$$

□

## 2.3 The algorithm terminates

The basic idea behind entropy compression is that a string of random bits cannot be represented by a shorter string. We will consider the string of random bits used for the recolouring steps of FIX and show that as we run FIX we can record a file which allows us to recover those random bits.

Each time we call  $\text{FIX}(g, \sigma)$ , we record the name of  $g$  and the colours of the vertices that determine  $g$ . It is not hard to see that this, along with the current colouring, will allow us to reconstruct all of the preceding random colour choices. Because the colours which determine  $g$  indicate that something unlikely occurred (namely the flaw  $g$ ), we can represent those colours in a very concise way. However, it may take a large amount of space to record the name of  $g$ . So instead, we use the degree bound in our graph to record a concise piece of information that will allow us to determine the name of  $g$ . This will lead to a compression of those random colour choices if the algorithm continues for too many steps.

First we describe these concise representations. Consider any vertex  $v$ . Let  $N^3(v)$  denote the set of vertices within distance 3 of  $v$  (including  $v$  itself). For each  $1 \leq \ell \leq |N^3(v)| < \Delta^3$  we let  $\omega(\ell, v)$  denote the  $\ell$ th vertex of  $N^3(v)$  when those vertices are listed in order of their labels. When we call, eg.  $\text{FIX}(B_w, \sigma')$  while running  $\text{FIX}(Z_v, \sigma'')$ , rather than recording the name “ $B_w$ ” it will suffice to just record “ $(B, \ell)$ ” where  $w = \omega(\ell, v)$ . So despite the fact that the number of vertices, and hence the size of the label of  $w$ , is not bounded in terms of  $\Delta$ , we are able to record  $w$  using only roughly  $3 \log_2 \Delta$  bits.

Suppose that we are given lists  $\mathcal{L} = \{L_u : u \in N_v\}$  of available colours for the neighbours of  $v$ . Let  $\mathcal{B}(\mathcal{L})$ , resp.  $\mathcal{Z}(\mathcal{L})$  be the set of all colour assignments from that list such that  $B_v$ , resp.  $Z_v$ , holds. Lemma 6 implies that  $|\mathcal{B}(\mathcal{L})|, |\mathcal{Z}(\mathcal{L})| < \Delta^{-4} \prod_{u \in N_v} |L_u|$ . For each  $1 \leq \ell \leq |\mathcal{B}(\mathcal{L})| + |\mathcal{Z}(\mathcal{L})|$ , we let  $\beta(\ell, \mathcal{L})$  denote the  $\ell$ th member of  $\mathcal{B}(\mathcal{L}) \cup \mathcal{Z}(\mathcal{L})$ . When we run, eg.  $\text{FIX}(B_v)$  we record the colours of  $N_v$  *before they get recoloured*; but instead of listing all the colours, we only need to record the value  $\ell$  such that those colours are  $\beta(\ell, \mathcal{L})$ .

We add some write statements to  $\text{FIX}$  as follows.

**FIX**( $f, \sigma$ )

Set  $\mathcal{L} = \{L_u : u \in N_{v(f)}\}$ .

Write “COLOURS =  $\ell$ ” where  $f = \beta(\ell, \mathcal{L})$ .

(\*) Set  $v = v(f)$  and assign each  $u \in N_v$  a uniformly selected colour from  $L_u$ .

While there are any flaws  $B_w$  with  $\text{dist}(w, v) \leq 3$  or  $Z_w$  with  $\text{dist}(w, v) \leq 2$ :

Let  $g$  be the least such flaw and call **FIX**( $g, \sigma'$ ) where  $\sigma'$  is the current colouring.

Write “FIX (B, $\ell$ )” or “FIX (Z, $\ell$ )” (depending on whether  $g$  is a B-flaw or an Z-flaw)  
where  $v(g) = \omega(v(f), \ell)$

Return the current colouring.

Write “Return”

Let  $\sigma_0$  be any initial colouring and let  $f$  be any flaw of  $\sigma_0$ . We will analyze a run of  $\text{FIX}(\sigma_0, f)$ . After  $t$  executions of the line (\*) we set

$\sigma_t$  is the current colouring

$H_t$  is the file that we write to

$R_t$  is the string of random bits that were used for all executions of (\*)

In our formal proofs, we will not in fact make use of  $R_t$ ; we only use it to give an intuitive picture of the compression of our random bits. Thus we are not careful about issues such as ensuring that each random choice uses an integer number of bits.



**Lemma 7** *Given  $\sigma_0, \sigma_t, f, H_t$  we can reconstruct the first  $t$  steps of FIX.*

**Proof** Let  $f_i$  denote the flaw addressed during the  $i$ th execution of  $(*)$ . First observe that  $f_1, \dots, f_t$  can be determined by  $\sigma_0, f, H_t$ . Indeed, proceed inductively: We know the sequence  $f_1 = f, \dots, f_{i-1}$ . FIX  $(f_i, \sigma_{i-1})$  was called while executing FIX  $(f_j, \sigma_{j-1})$  for some  $j < i$ . The locations of the “Return” lines in  $H_t$  are enough to determine the value of  $j$ , and by induction we know  $f_j$ . So the  $i$ th “FIX  $(-, \ell)$ ” line tells us that  $v(f_i) = \omega(v(f_j), \ell)$  and also tells us whether  $f_i = B_{v(f_i)}$  or  $Z_{v(f_i)}$ .

Next observe that, having determined  $f_1, \dots, f_t$ , we can reconstruct the colours assigned in each execution of  $(*)$  from  $H_t$  and  $\sigma_t$ . To see this, note that we can reconstruct  $\sigma_{t-1}$  from  $H_t, \sigma_t, f_t$ . We know that  $\sigma_{t-1} = \sigma_t$  on all vertices other than  $N_{v(f_t)}$ . This and the fact that our graph is triangle-free imply that for every  $u \in N_v$ , the list  $L_u$  does not change during step  $t$ . So  $\sigma_t$  tells us what  $\mathcal{L} = \{L_u : u \in N(v(f_t))\}$  was before the  $t$ th execution of  $(*)$ . Thus the  $t$ th “COLOURS= $\ell$ ” line tells us that  $\sigma_{t-1}(N_{v(f_t)}) = \beta(\ell, \mathcal{L})$ . Furthermore,  $\mathcal{L}$  and  $\sigma_t(N_v)$  tell us what colours were selected during the  $t$ th execution of  $(*)$ . Working backwards, this determines  $\sigma_t, \sigma_{t-1}, \dots, \sigma_1$  and hence all of our random choices.  $\square$

So  $R_t$  can be represented by  $(\sigma_0, \sigma_t, f, H_t)$ . Next we show that if FIX  $(\sigma_0, f)$  continues for  $t$  steps, where  $t$  is large, then  $(\sigma_0, \sigma_t, f, H_t)$  when expressed in binary will be much shorter than  $R_t$ . Any method to represent a random string of bits by a much shorter string must fail w.h.p. So this implies that w.h.p. we terminate before very many steps.

After  $t$  executions of  $(*)$  we have recorded fewer than  $t$  “Return” lines, each requiring  $O(1)$  bits. We have recorded  $t - 1$  “FIX $(-, \ell)$ ” lines, each requiring  $O(1) + 3 \log_2 \Delta$  bits since we have  $\ell < \Delta^3$  in each such line. We have  $t$  “COLOURS =  $\ell$ ” lines; we set  $\Lambda_i$  to be the value of  $\prod_{u \in N(v(f_i))} |L_u|$  during the  $i$ th execution of this line. During that execution we have  $\ell \leq |\mathcal{B}(\mathcal{L})| + |\mathcal{Z}(\mathcal{L})| < 2\Delta^{-4}\Lambda_i$  and so we require  $\log_2 \Lambda_i - 4 \log_2 \Delta + O(1)$  bits to record “COLOURS =  $\ell$ ”. Each such line is followed by an execution of  $(*)$  in which the relevant sets  $L_u$  have not changed and so we use  $\log_2(\Lambda_i)$  random bits of  $R_t$ .

The number of choices for a partial list colouring is at most  $(q + 1)^n$  and so  $\sigma_0, \sigma_t$  can each be represented with at most  $n \log_2(q + 1) < n \log_2 \Delta$  bits. We express  $f$  by naming  $v(f)$  and whether  $f$  is  $B_v$  or  $Z_v$ ; this requires  $1 + \log_2 n$  bits. This yields:

$$|R_t| = \sum_{i=1}^t \log_2(\Lambda_i)$$

$$|(\sigma_0, \sigma_t, f, H_t)| = 2n \log_2 \Delta + \log n + 3(t - 1) \log_2 \Delta + \sum_{i=1}^t (\log_2(\Lambda_i) - 4t \log_2 \Delta + O(1)) \quad (2)$$

Thus  $|(\sigma_0, \sigma_t, f, H_t)| \leq |R_t| - \frac{1}{2}t \log_2 \Delta + 3n \log_2 \Delta$  which is much smaller than  $|R_t|$  if  $t$  is large in terms of  $n$ .

Annoying technical issues arise when  $\Lambda_i$  is not a power of 2, and so our formal proof focuses directly on  $\Lambda_i$  without actually producing the bitstream. We are also able to drop  $\sigma_0, f$  from  $(\sigma_0, \sigma_t, f, H_t)$ .

**Lemma 8** *For any partial colouring  $\sigma$  and any flaw  $f$  of  $\sigma$ , the probability that  $\text{FIX}(f, \sigma)$  continues for at least  $3n$  executions of (\*) is at most  $\Delta^{-n/2}$ .*

**Proof** Set  $T = 3n$  and run  $\text{FIX}(f, \sigma)$  until it either terminates or carries out  $T$  executions of (\*).

Let  $\mathcal{Q}$  be any possible run of  $\text{FIX}(f, \sigma)$  that lasts for at least  $T$  executions. At the  $i$ th execution, recall that  $\Lambda_i = \prod_{u \in N_v} |L_u|$ . We choose our colouring by taking a uniform integer  $x_i$  from  $\{1, \dots, \Lambda_i\}$ . Note that  $\Lambda_i$  is determined by  $f, \sigma$  and  $x_1, \dots, x_{i-1}$ . Set  $\Lambda = \Lambda(\mathcal{Q}) = \prod_{i=1}^T \Lambda_i$  and set  $\lambda = \lambda(\mathcal{Q}) = \lfloor \log_2 \Lambda \rfloor$ . The probability that we carry out the run  $\mathcal{Q}$  is  $1/\Lambda \leq 2^{-\lambda}$ .

Let  $H_T = H_T(\mathcal{Q})$  be the history file created during the run of  $\mathcal{Q}$ , expressed in binary. The same calculations as in (2) yield:

$$|(H_T, \sigma_T)| \leq \lambda - T \log_2 \Delta + n \log_2 \Delta + O(1) \leq \lambda - n \log_2 \Delta.$$

So for each value of  $\lambda$ , the number of choices for  $H_T$ , and hence the number of choices for  $\mathcal{Q}$  with  $\lambda(\mathcal{Q}) = \lambda$ , is at most  $2^\lambda \times \Delta^{-n}$ .

At each step, we have  $|L_u| \leq q + 1 < \Delta$  for all  $u \in N_v$  and so  $\Lambda_i < \Delta^\Delta$ . It follows that  $\lambda \leq T \times \Delta \log \Delta$ . So summing over all choices of  $\lambda$ , we have that the total probability of all possible runs  $\mathcal{Q}$  that last for at least  $T$  executions is at most:

$$\sum_{\lambda=1}^{T\Delta \log \Delta} 2^\lambda \times \Delta^{-n} \times 2^{-\lambda} = 3n\Delta \log \Delta \times \Delta^{-n} < \Delta^{-n/2},$$

for nontrivial  $n, \Delta$ . This proves the lemma.  $\square$

As described above, these provide a proof of Theorem 1:

**Proof of Theorem 1** Consider any  $\epsilon > 0$  and any assignment of lists of size  $q = (1+\epsilon)\Delta/\log \Delta$  colours to the vertices. We begin by assigning Blank to every vertex. Then we repeatedly call  $\text{FIX}$  to eliminate any remaining flaws. More formally: While there is any flaw  $f$  we call  $\text{FIX}(f, \sigma)$  where  $\sigma$  is the current partial colouring. By Lemma 8 each call terminates within  $O(n)$  executions of (\*) with probability at least  $1 - \Delta^{-n/2}$ . By Observation 5, the number of flaws decreases by at least one after each call. There are at most  $2n$  initial flaws and so we obtain a flaw-free partial colouring  $\sigma^*$  after at most  $2n$  calls of  $\text{FIX}(f, \sigma)$  with probability at least  $1 - 2n\Delta^{-n/2} > 0$ . Lemma 4 implies that the Blank vertices of  $\sigma^*$  can be recoloured to give the required proper list colouring.  $\square$

**Remark:** This easily yields a polytime algorithm to produce the list colouring. Calling  $\text{FIX}$  at most  $2n$  times w.h.p. produces  $\sigma^*$  in  $O(n^2\Delta^2 \log \Delta)$  time; in fact, extending the definition of  $H_t, R_t, \sigma_t$  to cover the sequence of colourings/executions produced over the sequence of at most  $2n$  calls of  $\text{FIX}$  can reduce this running time to  $O(n \log n \Delta^2 \log \Delta)$  (see eg. the approach in [1]). The main result of [1] yields a polytime algorithm corresponding to Lemma 4, which we use to complete the colouring.

### 3 $K_r$ -free graphs

With a more complicated recolouring step, the same proof can be adapted to  $K_r$ -free graphs. The setup is the same as in Section 2 except with a larger list size:

We have a set of colours  $\mathcal{C}$ . Each vertex  $v$  has a list  $\mathcal{L}_v \subseteq \mathcal{C}$  of size

$$q = 200r \frac{\Delta \ln \ln \Delta}{\ln \Delta}.$$

We add the colour Blank to each list. A partial list colouring  $\sigma$  is an assignment to a subset of the vertices, where the colours are drawn from their lists. Given any partial colouring,  $L_v \subseteq \mathcal{L}_v$  is defined to be the set of colours not appearing on any neighbours of  $v$ .

Because we are not trying to get a good constant, we can afford to be a bit looser in our definition of  $L$  and our second flaw will be simpler than that in Section 2. We define

$$L = \Delta^{9/10}.$$

Given a partial colouring  $\sigma$ , we define the following two *flaws* for any vertex  $v$ :

$$\begin{aligned} B_v &\equiv L_v < L \\ Z_v &\equiv \text{at least } L \text{ neighbours of } v \text{ are coloured Blank.} \end{aligned}$$

It is trivial to see that any flaw-free partial colouring can be completed greedily to a full colouring of  $G$ , as the list of available colours for each vertex is greater than the number of uncoloured neighbours.

Again, we say  $v$  is the *vertex* of flaw  $f = B_v$  or  $Z_v$ , and we denote  $v(f) := v$ . We use the same ordering on the flaws: Every  $B_v$  comes before every  $Z_u$ , and the  $B_v$ 's and  $Z_u$ 's are each ordered according to the labels of  $v, u$ .

We find a flaw-free partial colouring using essentially the same algorithm we used for triangle-free graphs, but we must be more careful about recolouring a neighbourhood. It will be useful to represent a partial colouring of a neighbourhood as a collection of disjoint independent sets:

**Definition 9** *For each vertex  $v$ , a colour assignment to  $N_v$  is a collection of disjoint independent sets  $(\theta_1, \dots, \theta_{|\mathcal{C}|})$ , each a subset of  $N_v$ , such that for any  $u \in \theta_i$  we have:  $i \in \mathcal{L}_u$  and  $i$  does not appear on any neighbour of  $u$  outside of  $N_v$ .*

It is possible for  $\theta_i = \emptyset$ , and we do not require that  $\bigcup_{i=1}^{|\mathcal{C}|} \theta_i = N_v$ . Any  $u \in N_v$  that is not in any of the  $\theta_i$  is considered to be coloured Blank.

To recolour  $N_v$ , we take a uniformly random colour assignment to  $N_v$  and then assign the colour  $i$  to every vertex in each  $\theta_i$ . More specifically, given a colouring  $\sigma$  and a vertex  $v$ , we let  $\Omega$  denote the set of all colour assignments to  $N_v$  and we choose a uniform member of  $\Omega$ .

Note that if  $N_v$  contains no edges, then this recolouring is equivalent to giving each  $u \in N_v$  a uniform colour from  $N_u$ , as we did in FIX.

**FIX2**( $f, \sigma$ )

Set  $v = v(f)$ .

Choose a uniformly random colour assignment to  $N_v$  and then recolour  $N_v$  accordingly.

While there are any flaws  $B_w$  with  $\text{dist}(w, v) \leq 3$  or  $Z_w$  with  $\text{dist}(w, v) \leq 2$ :

Let  $g$  be the least such flaw and call **FIX**( $g, \sigma'$ ) where  $\sigma'$  is the current colouring.

Return the current colouring.

The analog of Observation 5 holds again here, and so to prove Theorem 2 it suffices to prove that **FIX2** terminates with positive probability.

We will assume throughout the remainder of this section that  $\Delta \geq 2^{200r}$  as otherwise the bound of Theorem 2 is trivial.

### 3.1 More probability bounds

We begin with some key lemmas from Shearer's bound on the independence number of a  $K_r$ -free graph[27]. We rephrase the short proofs here for completeness and to extract a useful fact from them.

Given a graph  $H$ , we define:

$I(H)$  is the number of independent sets of  $H$ .

**Lemma 10** *If  $H$  is  $K_r$ -free then  $2^{|V(H)|} \geq I(H) \geq 2^{|V(H)|^{\frac{1}{r-1}} - 1}$ .*

**Proof** The upper bound is trivial, as this is the number of subsets of  $V(H)$ . For the lower bound, we will prove that  $H$  has an independent set of size at least  $|V(H)|^{1/r-1} - 1$ ; the bound follows by considering all subsets of that independent set.

We proceed by induction on  $r$ . The trivial base case is  $r = 2$ . For  $r \geq 3$ : If any vertex  $u \in H$  has degree at least  $d = |V(H)|^{\frac{r-2}{r-1}}$  then we obtain a sufficiently large independent set in  $N_u$  by induction, since  $N_u$  is  $K_{r-1}$ -free. Otherwise, the straightforward greedy algorithm finds an independent set of size at least  $|V(H)|/(d+1) > |V(H)|^{1/r-1} - 1$ .  $\square$

**Lemma 11** *If  $H \neq \emptyset$  is  $K_r$ -free,  $r \geq 4$ , then half of the independent sets in  $H$  have size at least  $\frac{1}{2r} \log_2 I(H) / \log_2 \log_2 I(H)$ .*

**Proof** It suffices to show that at most  $\frac{1}{2}I(H)$  subsets of  $V(H)$  have size at most  $\ell = \lfloor \frac{1}{2r} \log_2 I(H) / \log_2 \log_2 I(H) \rfloor$ ; i.e:

$$\sum_{i=0}^{\ell} \binom{|V(H)|}{i} \leq \frac{1}{2}I(H). \quad (3)$$

We can assume  $\log_2 I(H) \geq 2$  as otherwise  $\ell = 0$  and so the lemma is trivial (since  $H \neq \emptyset$ ). We can also assume  $r \leq \log_2 I(H) / 2 \log_2 \log_2 I(H)$  else  $\ell = 0$ . We set  $x = \log_2 I(H) \geq 2$ . Rearranging

the second inequality of Lemma 10 gives  $|V(H)| \leq (1 + \log_2 I(H))^{r-1}$  and so we substitute  $h = (1 + \log_2 I(H))^{r-1}$  for  $|V(H)|$  in (3). So  $h = (1 + x)^{r-1} < \frac{1}{4}x^{2r}$  for  $x \geq 2$ . So the LHS of (3) is at most

$$2h^\ell < \frac{1}{2}x^{2r\ell} \leq \frac{1}{2}2^{\log_2 x \times \frac{x}{\log_2 x}} = \frac{1}{2}2^x = \frac{1}{2}I(H).$$

This proves (3).  $\square$

**Remarks:**

(1) Lemma 10 is the only place where we use the fact that our graph is  $K_r$ -free. Our proof shows that the bound of Theorem 2 holds whenever every subgraph  $H \subseteq G$  satisfies the implication of either Lemma 10 or Lemma 11. In fact, it is enough for this to hold for every  $v$  and  $H \subseteq N(v)$ .

(2) Note that the argument in Lemma 11 can in fact show that the average size of the independent sets of  $H$  is at least  $\frac{1}{2r} \log I(H) / \log \log I(H)$ , which is Lemma 1 of [27].

(3) Alon [4] proves that if  $G$  is locally  $r$ -colourable, meaning that every neighbourhood can be  $r$ -coloured, then for any  $v$  and  $H \subseteq N_v$ , the median size of the independent sets of  $H$  is at least  $\frac{1}{10 \log(r+1)} \log I(H)$ . Plugging this bound into the rest of our proof proves that  $\chi_\ell \leq O(\log r \frac{\Delta}{\log \Delta})$  for such graphs, as shown in [15].

We use these to bound the probabilities of our flaws.

**Setup for Lemma 12:** Each vertex  $u \in N_v$  has a list  $L_u$  containing Blank and perhaps other colours. We give the vertices of  $N_v$  a random colour assignment consistent with these lists. This assignment determines  $L_v$  - the set of colours in  $\mathcal{L}_v$  that do not appear in the colour assignment.

**Lemma 12** (a)  $\Pr(|L_v| < L) < \Delta^{-4}$ .

(b) *The probability that at least  $L$  neighbours of  $v$  are coloured Blank and  $|L_u| > L$  for all  $u \in N_v$  is at most  $\Delta^{-4}$ .*

**Proof** We begin with a method for sampling a colour assignment.

Define  $\Omega$  to be the set of all colour assignments, and let  $W = (W_1, \dots, W_{|\mathcal{C}|})$  be a uniform member of  $\Omega$ . Define  $Q_1$  to be the vertex set consisting of  $W_1$  and all blank vertices which can be given the colour 1; i.e. all blank  $u \in N_v$  with  $1 \in L_u$  and 1 not appearing on any neighbour of  $u$  outside of  $N_v$ . Select a uniformly random independent set  $W'_1$  of  $Q_1$  and form  $W'$  by replacing  $W_1$  with  $W'_1$ .

*Claim 1:  $W'$  is a uniform member of  $\Omega$ .*

*Proof of Claim 1:* For any  $|\mathcal{C}| - 1$  disjoint independent sets  $S_2, \dots, S_{|\mathcal{C}|} \subseteq N_v$  we define  $\Omega_{S_2, \dots, S_{|\mathcal{C}|}} \subseteq \Omega$  to be the set of colour assignments  $(\theta_1, \dots, \theta_{|\mathcal{C}|})$  with  $\theta_2 = S_2, \dots, \theta_{|\mathcal{C}|} = S_{|\mathcal{C}|}$ ; so this yields a partition of  $\Omega$ . Note that  $W'$  is a uniform member of  $\Omega_{W_2, \dots, W_{|\mathcal{C}|}}$ . Furthermore, because  $W$  is a uniform member of  $\Omega$ , the part  $\Omega_{W_2, \dots, W_{|\mathcal{C}|}}$  is selected with the correct distribution, i.e with probability  $|\Omega_{W_2, \dots, W_{|\mathcal{C}|}}|/|\Omega|$ . So  $W'$  is a uniform member of  $\Omega$ .  $\square$

Repeating this argument, we can resample  $W_2, \dots, W_{|\mathcal{C}|}$  in the same manner. Specifically:

Let  $W = (W_1, \dots, W_{|\mathcal{C}|})$  be a uniform member of  $\Omega$ .

For  $i = 1$  to  $|\mathcal{C}|$

Define  $Q_i$  to be the subgraph induced by  $W_i$  and all blank vertices which can be given the colour  $i$ .

Let  $W'_i$  be a uniform independent set of  $Q_i$

Modify  $W$  by replacing  $W_i$  with  $W'_i$ .

Note that the blank vertices in the definition of  $Q_i$  are blank in  $W = (W'_1, \dots, W'_{i-1}, W_i, \dots, W_{|\mathcal{C}|})$ . By repeating the argument from Claim 1, we see that the colour assignment produced by this procedure is a uniform member of  $\Omega$ .

*Part (a):* Let  $A_1$  be the set of colours  $i \in \mathcal{L}_v$  such that  $I(Q_i) \leq \Delta^{1/20}$ , and let  $A_2$  be the colours  $\mathcal{L}_v \setminus A_1$ . Since  $N_v$  is  $K_{r-1}$ -free, Lemma 11 implies that for each  $i \in A_2$  the median independent set of  $Q_i$  has size at least  $\frac{1}{2(r-1)} \log I(Q_i) / \log \log I(Q_i) > \frac{1}{40r} \log \Delta / \log \log \Delta$ . (When applying Lemma 11 note that if  $Q_i = \emptyset$  then  $i \in A_1$ .)

For each colour  $i \in A_1$ , the probability that  $W'_i = \emptyset$  is  $\frac{1}{I(Q_i)} \geq \Delta^{-1/20}$ . Note that if  $W'_i = \emptyset$  then  $i$  will be in  $L_v$ . For each colour  $i \in A_2$ , with probability at least  $\frac{1}{2}$  we have  $|W'_i| \geq \frac{1}{40r} \log \Delta / \log \log \Delta$ . Since the total size of the sets  $W'_i$  is at most  $\Delta$ , this can't happen for more than  $40r \frac{\Delta \log \log \Delta}{\log \Delta}$  colours.

We consider two random binary strings, each of length  $\mathcal{C}$ . In the first, each bit is 1 with probability  $\Delta^{-1/20}$ , and 0 otherwise. In the second, the bits are uniform. By coupling the choice of  $W'_i$  with these bits, we ensure that: (a) for each  $i \in A_1$ , if the corresponding bit in the first stream is 1 then  $W'_i = \emptyset$ ; (b) for each  $i \in A_2$ , if the corresponding bit in the second stream is 1 then  $|W'_i| \geq \frac{1}{40r} \log \Delta / \log \log \Delta$ . For example, in iteration  $i$  if we have  $I(Q_i) < \Delta^{1/20}$  and so  $i \in A_1$  then we look at the next bit of the first string. If it is 1 then we set  $W'_i = \emptyset$ ; otherwise we set  $W'_i = \emptyset$  with probability  $\frac{1}{I(Q_i)} - \Delta^{-1/20}$ , and similarly when  $i \in A_2$ .

Suppose that  $|L_v| < L$ . Then either at most  $L$  of the first  $\ell = \frac{1}{2}|\mathcal{L}_v| = 100r\Delta \log \log \Delta / \log \Delta$  bits of the first string are 1 (if  $|A_1| \geq \frac{1}{2}|\mathcal{L}_v|$ ) or at most  $40r \frac{\Delta \log \log \Delta}{\log \Delta}$  of the first  $\ell$  bits of the second stream are 1 (if  $|A_2| \geq \frac{1}{2}|\mathcal{L}_v|$ ). Note that the expected number of 1's in the first  $\ell$  bits of the first string is  $\ell \times \Delta^{-1/20} \gg L = \Delta^{9/10}$  and the expected number of 1's in the first  $\ell$  bits of the second string is  $\frac{1}{2}\ell = 50r\Delta \log \log \Delta / \log \Delta$ . So Lemma 3 (or the Chernoff Bounds) implies that each of these two events happens with probability less than  $\frac{1}{2}\Delta^{-4}$  for  $r \geq 4$  and  $\Delta \geq 2^{200r}$ . This proves part (a).

*Part (b):* Consider any  $L$  neighbours  $u_1, \dots, u_L \in N_v$ . We will prove the probability that each  $u_i$  is coloured blank and satisfies  $|L_{u_i}| > L$  is at most  $1/L!$ . This proves part (b) as  $\binom{\Delta}{L}/L! < \Delta^{-4}$  for  $\Delta \geq 2^{200r}$ .

Let  $\Omega_B \subset \Omega$  be the set of colour assignments in which every  $u_i$  is coloured Blank and satisfies  $|L_{u_i}| > L$ . (Note: a colour assignment in  $\Omega_B$  may also have additional blank vertices.) Take any  $W \in \Omega_B$  and extend it to a colour assignment  $W_2$  in which each of  $u_1, \dots, u_t$  are not blank as follows:

begin with the colouring  $W$

for  $i = 1$  to  $L$

give  $u_i$  a colour from  $\mathcal{L}_{u_i}$  which does not appear on any of its neighbours

This yields a colouring  $W'$  of  $N_v$  which can be viewed as the colour assignment  $(\theta_1, \dots, \theta_{|\mathcal{C}|})$  where  $\theta_j$  is the set of vertices with colour  $j$  in  $W'$ .

By definition of  $\Omega_B$ , each  $u_i$  has at least  $L$  available colours in  $W$ . By the time we reach iteration  $i$ , at most  $i - 1$  of those colours have been assigned to a neighbour of  $u_i$  in  $\{u_1, \dots, u_{i-1}\}$ . So there are always at least  $L - i + 1$  choices for a colour to assign to  $u_i$  and so the number of choices for  $W'$  is at least  $L!$ . Each colour assignment  $W'$  can arise from at most one  $W \in \Omega_B$ , namely the  $W$  obtained from  $W'$  by colouring  $u_1, \dots, u_L$  all Blank. So  $|\Omega_B| \leq |\Omega|/L!$ , which is what we need to establish part (b).  $\square$

### 3.2 FIX2 terminates

Now the same argument from Section 2.3 implies that FIX2 terminates with positive probability, and thus proves Theorem 2.

Each time we call FIX2  $(v, \sigma)$  we let  $\mathcal{L} = \{L_u : u \in N_v\}$  be the lists of available colours on the neighbours of  $v$  in the colouring obtained from  $\sigma$  by uncolouring  $N_v$ ; i.e.  $L_u$  is the set of colours in  $\mathcal{L}_u$  that do not appear on any neighbours of  $u$  outside of  $N_v$ . We let  $\Omega(\mathcal{L})$  be the set of colour assignments to  $N_v$  consistent with  $\mathcal{L}$ . We let  $\mathcal{B}(\mathcal{L}) \subset \Omega(\mathcal{L})$  be the set of colour assignments that have the flaw  $B_v$ . We let  $\mathcal{Z}(\mathcal{L}) \subset \Omega(\mathcal{L})$  be the set of colourings which have the flaw  $Z_v$ .

We define  $H_t, R_t$  analogously to Section 2.3. At each step: If we are addressing the flaw  $B_v$  then Lemma 12(a) implies that the number of choices for the colouring of  $N_v$  before the recolour line is at most  $|\mathcal{B}(\mathcal{L})| \leq \Delta^{-4}|\Omega(\mathcal{L})|$ . If we are addressing the flaw  $Z_v$  then by our ordering of the flaws, each  $u \in N_v$  has at least  $L$  available colours in  $\sigma$  and so must have  $|L_u| \geq L$  after uncolouring  $N_v$ . So Lemma 12(b) implies that the number of choices for the colouring of  $N_v$  before the recolour line is at most  $|\mathcal{Z}(\mathcal{L})| \leq \Delta^{-4}|\Omega(\mathcal{L})|$ . This yields that the size of what is written to  $H_t$  is  $3 \log_2 \Delta + \log_2 |\Omega(\mathcal{L})| - 4 \log_2 \Delta + O(1)$  whereas the number of random bits used is  $\log_2 |\Omega(\mathcal{L})|$ . This is enough for the analysis from Section 2.3 to carry through.

**Remark** This time it is not clear how to obtain a polytime algorithm; the challenge is to select a uniform colour assignment efficiently. Johansson's proof yields a polytime algorithm (see [6]).

## Acknowledgement

My thanks to Dimitris Achlioptas and Fotis Iliopoulos for some very helpful discussions. This research is supported by an NSERC Discovery grant.

## References

- [1] D. Achlioptas and F. Iliopoulos. *Random walks that find perfect objects and the Lovasz Local Lemma* JACM (to appear). Preliminary version in Proc. of FOCS (2014).
- [2] D. Achlioptas and F. Iliopoulos. *Focused local search and the Lovasz Local Lemma*. Proc. of SODA (2016).
- [3] M. Ajtai, P. Erdos, J. Komlos and E. Szemerédi. *On Turans theorem for sparse graphs*. Combinatorica **1** (1981), 313 - 317.

- [4] N. Alon. *Independence numbers of locally sparse graphs and a ramsey type problem*. Rand. Str. & Alg. **9** (1996) 271 - 278.
- [5] N. Alon, M. Krivelevich, B. Sudakov. *Coloring graphs with sparse neighborhoods*. J. Comb. Th. B **77** (1999), 73 - 82.
- [6] N. Bansal, A. Gupta and G. Guruganesh. *On the Lovász Theta function for independent sets in sparse graphs*. Proceedings of STOC 2015.
- [7] B. Bartłomiej, S. Czerwinski, J. Grytczuk, and P. Rzazewski. *Harmonious coloring of uniform hypergraphs*. Applicable Anal. and Disc. Math. **10** (2016), 73 - 87.
- [8] V. Dujmovic, G. Joret, J. Kozik, and D.R. Wood. *Nonrepetitive colouring via entropy compression*. Combinatorica **36** (2016), 661 - 686.
- [9] P. Erdős and L. Lovász. *Problems and Results on 3-Chromatic Hypergraphs and Some Related Questions*. In: "Infinite and Finite Sets" (A. Hajnal et. al. Eds), Colloq. Math. Soc. J. Bolyai **11**, North Holland, Amsterdam (1975), 609 - 627.
- [10] L. Esperet and A. Parreau. *Acyclic edge-coloring using entropy compression*. European J. Comb. **34** (2013), 1019 - 1027.
- [11] L. Fortnow. *A Kolmogorov complexity proof of the Lovász Local Lemma*. Blog post. <http://blog.computationalcomplexity.org/2009/06/kolmogorov-complexity-proof-of-lov.html>
- [12] A. Gagol, G. Joret, J. Kozik, and P. Micek. *Pathwidth and nonrepetitive list coloring*. Elec. J. of Comb. **23** (4):P4.40, (2016).
- [13] J. Grytczuk, J. Kozik, and P. Micek. *A new approach to nonrepetitive sequences*. Rand. Str. & Alg. **42** (2013), 214 - 225.
- [14] A. Johansson, *Asymptotic choice number for triangle free graphs*. Unpublished manuscript (1996).
- [15] A. Johansson, *The choice number of sparse graphs*. Unpublished manuscript (1996).
- [16] J.H. Kim, *On Brooks' Theorem for sparse graphs*, Comb. Prob. & Comp. **4** (1995), 97 - 132.
- [17] C. McDiarmid. *Concentration*. In: Probabilistic Methods for Algorithmic Discrete Mathematics, (Habib M., McDiarmid C., Ramirez-Alfonsin J., Reed B., Eds.), Springer (1998), 195 - 248.
- [18] M. Molloy and B. Reed. *Graph Colouring and the Probabilistic Method*. Springer (2002).
- [19] R. Moser. *A constructive proof of the Lovász Local Lemma*. Proceedings of the 41st ACM Symposium on Theory of Computing (2009).
- [20] R. Moser and G. Tardos. *A constructive proof of the general Lovász Local Lemma*. J. ACM, **57(2)** (2010).
- [21] A. Panconesi and A. Srinivasan *Randomized distributed edge coloring via an extension of the Chernoff-Hoeffding Bounds*. SIAM J. Comput., **26** (1997), 350 - 368.



- [22] S. Pettie and H. Su. *Distributed coloring algorithms for triangle-free graphs*. Information and Computation **243** (2015), 263 - 280.
- [23] J. Przybylo. *On the facial Thue choice index via entropy compression*. J. Graph Th. **77** (2014), 180 - 189.
- [24] J. Przybylo, J. Schreyer, E. Skrabulakova. *On the facial Thue choice number of plane graphs via entropy compression method*. Graphs and Combinatorics **32** (2016), 1137 - 1153.
- [25] B. Reed. *The list colouring constants*. J. Graph Th. **31** (1999), 149 - 153.
- [26] J. Shearer. *A note on the independence number of triangle-free graphs*. Discrete Math. **46** (1983), 83 - 87.
- [27] J. Shearer. *On the independence number of sparse graphs*. Rand. Str. & Alg. **7** (1995), 269 - 271.
- [28] T. Tao. *Moser's entropy compression argument*. Blog post.  
<https://terrytao.wordpress.com/2009/08/05/mosers-entropy-compression-argument/>